SPR 1/07/07

**AMENDMENTS TO THE SPECIFICATION**

Please amend the paragraph beginning on page 7, line 25 to page 8, line 3 of the specification as follows:

FIG. 7 is a flow chart showing operation of the receiver of FIG. 3 in accordance with the preferred embodiment of the present invention. The logic flow begins at step 701 where receiver 306 receives a frame. At step 703, logic unit 309 stores the current frame in a buffer (not shown), and at step 705, logic unit determines if a frame was received out of order. If, at step 705, logic unit 309 determines that a frame has been received out of order, then the logic flow continues to step 707, where a NAK is sent for the missing frame, otherwise the logic flow continues to step 709. At step 709, logic unit 309 determines if the received frame was an idle frame, and if so, the logic flow continues to step 711 where an idle-frame ACK is transmitted. As discussed above, the idle-frame acknowledgment indicates to the transmitter that an idle frame has been received, causing the transmitter to cease transmitting idle frames. If at step 709, logic unit 309 determines that an idle frame has not been received, the logic flow returns to step ~~307~~ 701.

have properly received an idle frame with sequence number 5 (I5). Therefore, after retransmission of the frame, no more idle frames are sent. Because transmitting circuitry 301 does not continue to send idle frames after the last data frame is NAK'd, fewer idle frames are transmitted to receiving circuitry 302, resulting in less system interference.

Receiver sends an idle-frame acknowledgment (Idle_ACK) when an idle frame is received

In the preferred embodiment of the present invention an Idle Acknowledgment (ACK) is transmitted when an idle frame has been received with a sequence number equal to the sequence number of the next new data frame expected. Upon reception of the Idle_ACK, transmitting circuitry 301 will cease sending idle frames to the receiving circuitry 302. This is illustrated in FIG. 6. As shown, once receiving circuitry 302 receives an idle frame with the next expected sequence number, an idle ACK is transmitted to transmitting circuitry. The idle ACK indicates to transmitting circuitry 301 that an idle frame has been received by receiving circuitry 302 with the corresponding sequence number. Transmitting circuitry immediately ceases sending any further idle frames if the Idle_Ack's sequence number is equal to the next sequence number to be sent. Because transmitting circuitry 301 does not continue to send idle frames after receiving an idle ACK, fewer idle frames are transmitted to receiving circuitry 302, resulting in less system interference.

FIG. 7 is a flow chart showing operation of the receiver of FIG. 3 in accordance with the preferred embodiment of the present invention. The logic flow begins at step 701 where receiver 306 receives a frame. At step 703, logic unit 309 stores the current frame in a buffer (not shown), and at step 705, logic unit determines if a frame was received out of order. If, at step 705, logic unit 309 determines that a frame has been received out of order, then the logic flow continues to step 707, where a NAK is sent for the missing frame, otherwise the logic flow continues to step 709. At step 709, logic unit 309 determines if the received frame was an idle frame, and if so, the logic flow continues to step 711 where an idle-frame ACK is transmitted. As discussed above, the idle-frame acknowledgment indicates to the transmitter that an idle frame has been received,

causing the transmitter to cease transmitting idle frames. If at step 709, logic unit 309 determines that an idle frame has not been received, the logic flow returns to step 307.

As discussed above, because transmitting circuitry 301 does not continue to send idle frames after receiving an idle ACK, fewer idle frames are transmitted to receiving circuitry 302, resulting in less system interference.

FIG. 8 is a flow chart showing operation of the transmitter of FIG. 3 in accordance with a first preferred embodiment of the present invention. In accordance with the first preferred embodiment, retransmission of a data frame does not reset the RLP idle frame counter. The logic flow begins at step 801 where logic unit 304 determines if the last data frame has been transmitted. If, at step 801 it is determined that the last data frame has not been transmitted, then the logic flow continues to step 803 otherwise the logic flow continues to step 805 where the idle frame counter is initialized (set to zero). At step 803 logic unit 304 instructs transmitter 303 to transmit the data frame and the logic flow returns to step 801.

After initializing the idle frame counter (step 805) the logic flow continues to step 807 where logic unit 304 instructs transmitter 303 to transmit an idle frame to the receiving circuitry. As discussed above, the idle frame includes a frame number incremented from the last data frame transmitted. After transmitting the idle frame, logic unit 304 increments the idle frame counter (step 809). Next, logic unit 304 determines if a NAK has been received (step 811), and if so logic unit 304 instructs transmitter 303 to retransmit the NAK'd frame (Step 813). The logic flow then continues to step 815. If at step 811 it is determined that a NAK has not been received, then the logic flow continues to step 815 where it is determined if the number of idle frames transmitted is equal to a threshold (N). If, at step 815 it is determined that the number of frames transmitted is equal to N, then the logic flow ends at step 817, otherwise the logic flow returns to step 807.

As discussed above, because idle frame counter 308 is not reset after retransmission of frames, fewer idle frames are transmitted to receiving circuitry 302, resulting in less system interference.

FIG. 9 is a flow chart showing operation of the transmitter of FIG. 3 in accordance with a second preferred embodiment of the present invention. In accordance with the second preferred embodiment, idle frames cease being transmitted when the last or final data frame is NAK'd. The logic flow is similar